**SSN College of Engineering Department of Computer Science and Engineering**

**III year - UCS1512 – Microprocessors Lab**

**Floating point operations**

**Exp No:** 09

**Name:** Rahul Ram M

**Register Number:** 185001121

**Date:** 13/10/2020

**9a) Floating point addition:**

**Aim:**

To design 8086-program for floating point addition.

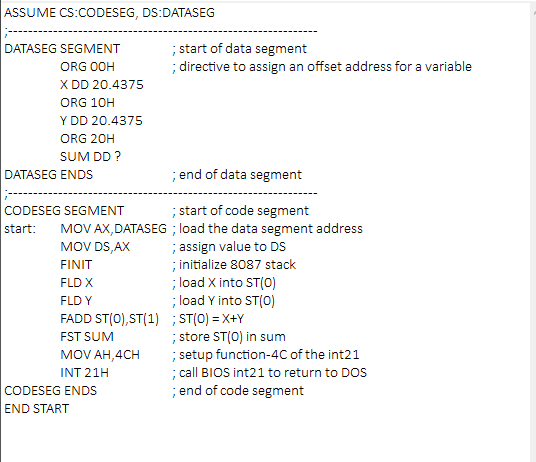
**Procedure for executing MASM:**

1. Run Dosbox and mount your masm folder to a drive in dosbox.
2. Goto the mounted drive.
3. Save the 8086 program with extension .asm in the same folder using command “edit”
4. After creating the file, assemble it using the command “masm filename.asm”
5. Link the file using the command “link filename.obj;”
6. Use debug command with filename.exe to execute and analyse the memory contents, “debug filename.exe”.
7. In debug, command “u” will display the unassembled code.
8. Use command “d segment:offset” to see the content of memory locations starting from segment:offset address.
9. To change the value in memory, use the command “e segment:offset”
10. Verify the memory contents to ensure the updates (using command “d”).
11. . Execute using the command “g” and check the outputs.
12. “q” to exit from debug and “exit” to exit from command prompt and to close the Dosbox.

**Algorithm:**

1. START: Move the starting address of data segment to AX register and move the data from AX register to DS register.
2. Initialize the 8087-stack using FINIT command. This stack will be used for floating point operations.
3. Load the floating-point number from variable X to the top of the stack i.e. ST (0) using FLD command.
4. Now again load the floating-point number from variable Y to the top of the stack i.e. ST (0) using FLD command. The previous stack top contents will be pushed into the stack.
5. Using FADD add ST(0) and ST(1) which stores the result in ST(0).
6. Using FST store the resulting floating-point number from the top of the stack to the variable SUM.
7. Move the hexadecimal value 4C into AH register. INT 21H means invoke the interrupt identified by the hexadecimal number 21. In MS-DOS, invoking interrupt 21h while AH = 4Ch causes the current process to terminate and uses the value of register AL as the exit code of the process.

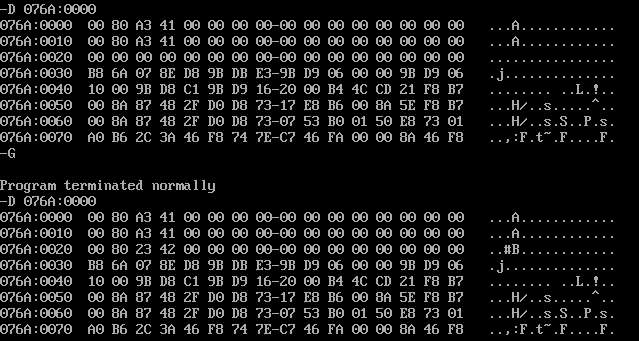
**Program:**



|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | MOV AX, DATA  MOV DS, AX | Transferring the data from DATA to AX register and  from AX register to DS register. |
| FINIT | Initialize 8087 stack. |
| FLD X | load X into ST(0) |
| FLD Y | load Y into ST(0) |
| FADD ST(0), ST(1) | ST(0) = X+Y |
| FST SUM | Store ST(0) in sum. |
| MOV AH,4CH | Setup function-4C of the int21. |
| INT 21H | Call BIOS int21 to return to DOS. |

**Unassembled Code:**

**Snapshot of sample input and output:**

**Result:**

ShapeThus the 8086 program for floating-point addition is executed successfully in DOS-BOX

**9b) Floating point subtraction:**

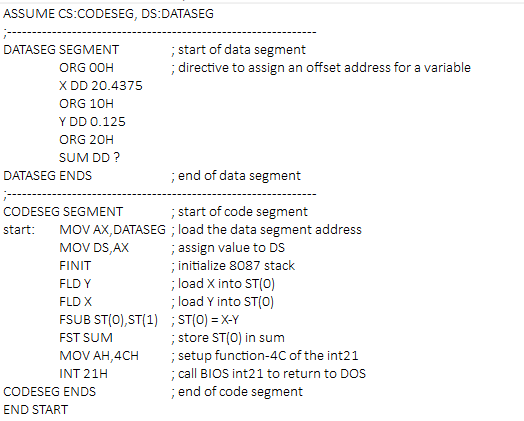
**Aim:**

To design 8086-program for floating point subtraction.

**Algorithm:**

1. START: Move the starting address of data segment to AX register and move the data from AX register to DS register.
2. Initialize the 8087-stack using FINIT command. This stack will be used for floating point operations.
3. Load the floating-point number from variable X to the top of the stack i.e. ST (0) using FLD command.
4. Now again load the floating-point number from variable Y to the top of the stack i.e. ST (0) using FLD command. The previous stack top contents will be pushed into the stack.
5. Using FSUB sub ST(0) from ST(1) and store the result in ST(0).
6. Using FST store the resulting floating-point number from the top of the stack to the variable SUM.
7. Move the hexadecimal value 4C into AH register. INT 21H means invoke the interrupt identified by the hexadecimal number 21. In MS-DOS, invoking interrupt 21h while AH = 4Ch causes the current process to terminate and uses the value of register AL as the exit code of the process.

**Program:**

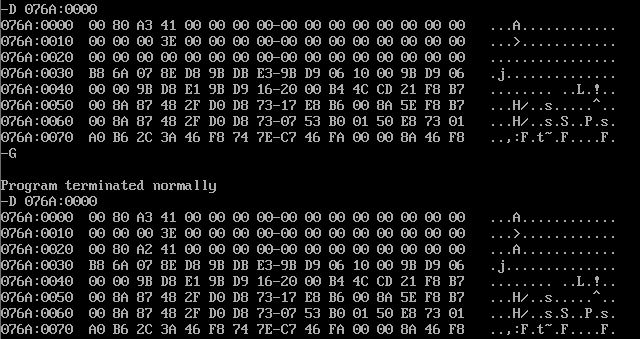


|  |  |  |
| --- | --- | --- |
|  | **Program** | **Comments** |
| START: | MOV AX, DATA  MOV DS, AX | Transferring the data from DATA to AX register and  from AX register to DS register. |
| FINIT | Initialize 8087 stack. |
| FLD X | load X into ST(0) |
| FLD Y | load Y into ST(0) |
| FSUB ST(0), ST(1) | ST(0) = X-Y |
| FST SUM | Store ST(0) in sum. |
| MOV AH,4CH | Setup function-4C of the int21. |
| INT 21H | Call BIOS int21 to return to DOS. |

**Unassembled Code:**



**Snapshot of sample input and output:**



**Result:**

ShapeThus the 8086-program for floating-point subtraction is executed successfully in DOS-BOX